

Please amend the application, without prejudice, as follows:

IN THE CLAIMS

Please cancel claims 4, 10, 16, and amend claims 1, 3, 7, 9, 13, and 15 to read as follows:

Sub C1  
1. A method for maintaining a security profile throughout nested service invocations on a distributed, component-based system, comprising the steps of:

- Sub C1  
B
- (a) providing interconnections between distributed components each having nested service invocations;
  - (b) identifying a user;
  - (c) associating the user with roles;
  - (d) creating a user context instance upon successful identification of the user, wherein the user context instance includes information about the user including the roles and a unique user identifier;
  - (e) receiving a request from the user to invoke a first service on a first component, wherein the first component invokes a second service of a second component such that the user context instance is passed as a parameter from the first component to the second component, and wherein completion of the second service is necessary to complete the first service;
  - (f) querying the user context instance for the unique user identifier;
  - (g) comparing the unique user identifier in the user context instance with an access control list for verifying that the user has access to the first component; and
  - (h) comparing the unique user identifier in the user context instance with an access control list for verifying that the user has access to the second service of the second component.

Sub C3  
3. A method as recited in claim 1, further comprising the step of modifying a user interface to provide access to actions that can be performed by the user based on the unique user identifier and the roles associated with the user.

Sub C6  
7. A computer program embodied on a computer readable medium for maintaining a security profile throughout nested service invocations on a distributed, component-based system, comprising:

- Sub C6  
B
- (a) a code segment that provides interconnections between distributed components each having nested service invocations;
  - (b) a code segment that identifies a user;
  - (c) a code segment that associates the user with roles;

- ~~3~~
- (d) a code segment that creates a user context instance upon successful identification of the user, wherein the user context instance includes information about the user including the roles and a unique user identifier;
  - (e) a code segment that receives a request from the user to invoke a first service on a first component, wherein the first component invokes a second service of a second component such that the user context instance is passed as a parameter from the first component to the second component, and wherein completion of the second service is necessary to complete the first service;
  - (f) a code segment that queries the user context instance for the unique user identifier;
  - (g) a code segment that compares the unique user identifier in the user context instance with an access control list for verifying that the user has access to the first component; and
  - (h) a code segment that compares the unique user identifier in the user context instance with an access control list for verifying that the user has access to the second service of the second component.

~~4~~  
~~Sub 4~~

9. A computer program as recited in claim 7, further comprising a code segment that modifies a user interface to provide access to actions that can be performed by the user based on the unique user identifier and the roles associated with the user.

~~5~~  
~~Sub C11~~

13. A system for maintaining a security profile throughout nested service invocations on a distributed, component-based system, comprising:

- (a) logic that provides interconnections between distributed components each having nested service invocations;
- (b) logic that identifies a user;
- (c) logic that associates the user with roles;
- (d) logic that creates a user context instance upon successful identification of the user, wherein the user context instance includes information about the user including the roles and a unique user identifier;
- (e) logic that receives a request from the user to invoke a first service on a first component, wherein the first component invokes a second service of a second component such that the user context instance is passed as a parameter from the first component to the second component, and wherein completion of the second service is necessary to complete the first service;
- (f) logic that queries the user context instance for the unique user identifier;
- (g) logic that compares the unique user identifier in the user context instance with an access control list for verifying that the user has access to the first component; and

~~5~~  
~~B~~

- (h) logic that compares the unique user identifier in the user context instance with an access control list for verifying that the user has access to the second service of the second component.
- 

~~16~~  
~~B~~  
Sub 13

15. A system as recited in claim 13, further comprising logic that modifies a user interface to provide access to actions that can be performed by the user based on the unique user identifier and the roles associated with the user.
-